

集成 ERP 的移动商业智能系统^①

祝锡永, 林宝川

(浙江理工大学 经济管理学院, 杭州 310018)

摘要: 集成 ERP 的商业智能虽然可以满足企业对海量数据进行智能分析与挖掘以辅助决策的需求, 但依然难以随时随地提供实时数据以快速响应市场变化和进行有效决策. 在现有商业智能研究基础上, 提出了集成 ERP 的实时移动商业智能概念, 运用中间件数据集成技术和面向服务架构, 设计了一个扩展性强、低耦合和跨平台的移动商业智能系统体系架构, 利用 Android 在智能移动终端上开发了一个与服务器端 ERP 系统相集成的商业智能系统, 并通过实例对该系统在移动终端上的实现技术进行了详细的描述.

关键词: 移动商业智能; ERP; 智能移动终端; Android

Mobile Business Intelligence Systems Integrated with ERP Applications

ZHU Xi-Yong, LIN Bao-Chuan

(School of Economics and Management, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: Business intelligence systems integrated with ERP applications have met the requirements of businesses for the support of decision-making by intelligent data mining and data analysis, but they still cannot provide real-time analytical data on an on-demand basis to help businesses quickly respond to market changes. First, an architecture of a real-time mobile business intelligence system integrated with ERP applications is proposed, then middleware technology and service oriented architecture are used to design a scalable, low coupling, and cross-platform framework of a mobile business intelligence system. Finally, an example is presented to describe in detail the implementation of the system in the mobile terminals by Android.

Key words: mobile business intelligence; ERP; intelligent mobile terminals; Android

1 引言

商业智能(Business Intelligence, BI)已成为当前企业信息化的一个重要组成部分. BI 系统与 ERP 应用相集成能有效地对企业现有海量数据进行多维分析和挖掘, 为科学决策和预测提供依据^[1]. 近年来, 国内外学者对集成 ERP 的实时商业智能进行了广泛的研究. Richard Hackathon 提出了一个实时商业智能系统的价值模型, 指出 BI 价值会在事件发生和企业做出响应的时段上流失, 从而使企业错过潜在的商机^[2]; 邵亮等提出了一种实时商业智能框架, 并与企业已有的应用系统(如 ERP、CRM 等)等进行集成, 使商业智能与企业应用系统之间具有无缝连接性^[3].

然而, 现有 BI 系统大多基于有线网络和计算机终端, 受地理位置限制, 仅适用于固定办公的应用场景. 随着新一代移动通信技术和智能移动终端的普及, 越来越多的企业希望借助智能移动终端设备能随时随地地访问应用系统中的数据, 并通过 BI 工具对数据进行联机分析处理后, 实时、动态地传送和发布到各种应用场景, 以便管理决策人员及时做出响应和控制^[4].

移动商业智能(Mobile Business Intelligence, MBI)借助新一代移动通信技术与手段, 将 BI 的实时性和集成性融于一体. 本文针对商业智能与智能移动设备的应用现状, 通过对传统 BI 系统架构进行改造, 提出一个具有可扩展性、低耦合性和跨平台特征的移

^① 基金项目:国家自然科学基金(71071144)

收稿时间:2012-02-16;收到修改稿时间:2012-03-30

动商业智能系统体系框架。

2 集成ERP的移动商业智能系统架构

2.1 MBI 系统的总体架构

移动商业智能具有以下主要特点^[5]：①集成性：与服务器端的ERP系统和BI分析工具相集成，由服务器端提供原始数据和联机分析结果；②移动性：借助无线通信技术和移动终端操作系统(如 Android, IOs 等)，在移动终端(如手机、平板电脑等)通过各种形式展示 BI 分析结果；③轻量级：由于移动终端的数据处理能力依然有限，大部分数据分析挖掘任务仍需由服务器端承担，通过与服务器端的即时通信交互，将服务器端已处理完成的信息在智能移动终端上直接输出，这样可以显著减轻移动终端的负荷；④安全性：对数据传输进行加密处理，保护数据安全。

本文给出一个 MBI 系统的总体架构(如图 1 所示)，它由数据层、业务逻辑层、中间层、Web 服务层和客户端层等组成，各层之间相互分离，可增强系统的可扩展性，有利于实现群组服务和负载均衡^[6]。

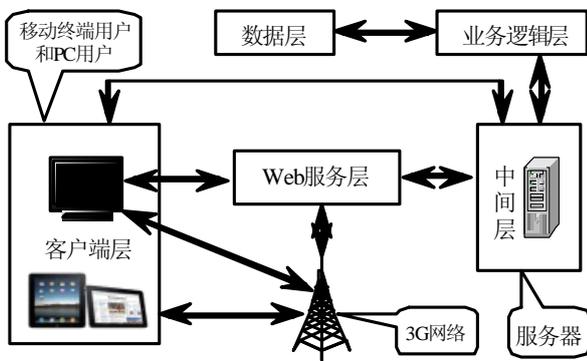


图 1 系统总体架构图

2.2 MBI 系统的数据层

MBI 数据层(如图 2 所示)包含系统所有数据资源。ERP 系统中的海量数据和企业外部数据借助中间件技术经过 ETL(抽取、清洗、转换、加载)等过程，转换为分析型数据，以元数据形式载入数据仓库。数据仓库中的数据根据不同需求，再经 ETL 工具装入到不同的数据集市，成为企业不同部门经营、管理和决策的数据库。对于实时性要求较高的 BI 数据，可通过元数据管理直接发布到移动终端^[7]。

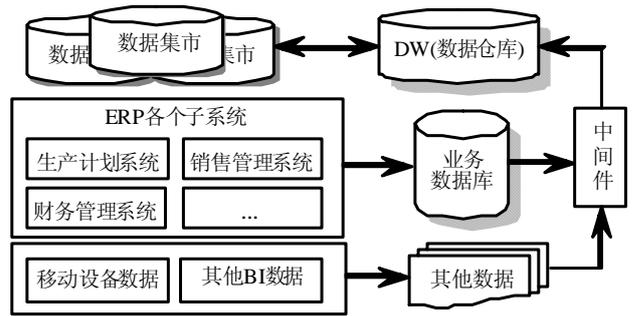


图 2 MBI 系统的数据层

数据层还可以通过第三方 API 与其他 BI 数据库、移动设备数据进行集成，同时可以与其他层相分离，以提高系统数据的安全性。

2.3 MBI 系统的业务逻辑层

业务逻辑层是 MBI 系统的核心，负责对整个系统业务逻辑进行控制，其功能模块如图 3 所示。业务逻辑层的设计遵循面向服务架构理论，根据特定的 Web 服务请求，将功能模块进行分离，使这些模块能够通过 Web 服务进行互相连接或者被独立地调用，以降低模块之间的耦合性。

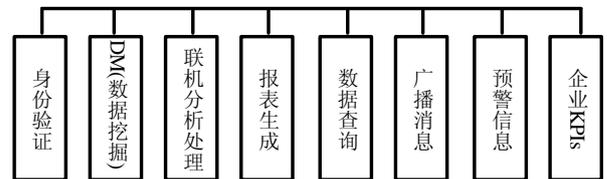


图 3 MBI 的业务逻辑层功能模块

业务逻辑层与系统数据层和 Web 服务层密切相关。例如，在报表生成业务逻辑中，通过对与客户端交互的报表 Web 服务进行访问，从请求的服务中获取参数后调用数据库连接，并将参数传递给数据层进行操作，再由数据层返回相关的数据，然后将数据填充到已设计好的图表或者报表中，并转换成适合的网页文件形式，最后将网页文件的地址返回给客户端所请求的报表 Web 服务^[8]。

2.4 MBI 系统的 Web 服务层

Web 服务层负责客户端和业务逻辑层之间的通信，传递用户输入的参数，并调用业务逻辑层中功能模块^[9]。以上述 Web 报表服务为例，该服务包含业务逻辑层中的数据查询和报表生成两个功能模块。根据查询条件，调用数据查询模块，获取目标信息后返回

给报表生成模块, 报表生成模块生成相应的报表, 并返回特定的 URL 给请求的用户, 其流程如图 4 所示。

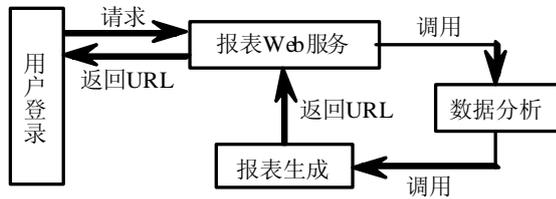


图 4 报表 Web 服务的实现流程

3 集成ERP的移动商业智能的实现

这里通过实例具体描述以手机为移动终端的 MBI 系统的关键技术。该实例使用 Web 服务、XML 及 Json 格式与移动终端进行数据传输, 将 BI 数据以仪表盘、计分卡以及图表等形式在移动终端进行展示。

3.1 技术平台的配置

数据资源以用友 ERP 数据库为例; ETL 采用 Kettle 工具及其插件, 将 ERP 系统中的数据进行定时实时的收集、转换、装载, 并按不同管理决策需求装载到不同的数据集中; 中间层(Web 服务器)采用 Apache Tomcat 6.0; Web 服务层的服务器端采用 Axis2 开发和发布 Web 服务; Web 服务客户端类库采用适合手机的 KSOAP2; 移动终端选取 Android 作为开发平台。

图 5 是一个模拟器环境下调用相应 Web 服务接口后显示的年度销售分析 BI 页面视图, 它由 Android 系统在手机移动终端上开发实现。

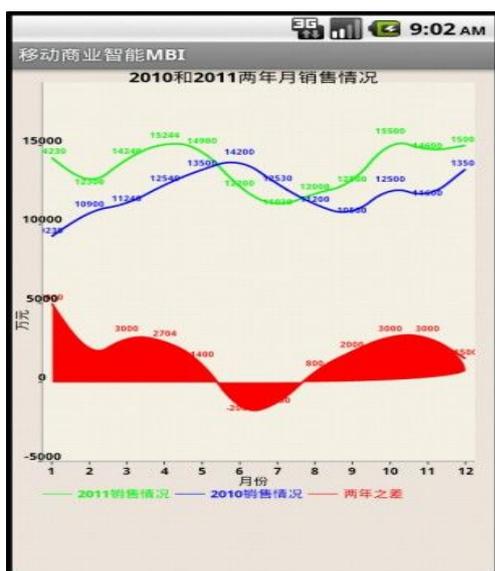


图 5 MBI 在手机终端上实现效果图

3.2 关键技术及其实现过程

(1) 服务器端部署和发布 Web 服务

安装 Apache Axis2。目前主要有标准二进制发行包、WAR 发行包、源码发行包和文档发行包 4 类[10]。本例选择 WAR 发行包, 将它解压到 Apache Tomcat 6.0 服务器的 Webapps 目录下。

编写 Web 服务应用程序, 即根据业务需求创建服务程序, 通过 Eclipse 的插件进行打包服务, 把打包好的服务部署到 Axis2 下, 至此 Web 服务发布成功。

(2) 移动终端异步调用 Web 服务

虽然计算机客户端的 Web 服务类库非常丰富, 但对于移动终端系统来说过于庞大, 因此使用面向移动设备的 KSOAP2 类库调用 Web 服务。KSOAP2 调用 Web 服务可分为下列主要步骤: 首先使用 SoapObject 对象指定调用的方法, 然后通过 HttpTransportSE 对象的 call 方法调用 Web 服务方法, 最后通过 getResponse 方法返回结果。当调用 Web 服务的用户很多时, 服务端响应会比较迟缓, 为此采用异步调用 Web 服务的方法来解决此现象, 其核心程序代码如下。

```

    public class MBIActivity extends Activity
    implements OnClickListener {class MyTask extends
    AsyncTask {
        @Override
        Protected Object doInBackground (Object... params) {try
        {private static final String SPACE = "http://MBIService";
        //定义命名空间
        private static final String METHOD = "MBISales";
        //定义调用的 WebService 中获取销售记录的方法名
        private static final String URL =
        "http://10.0.2.2:8080/
        Axis2/services/MBIAxis wsdl"; //服务器 WSDL 文档的 URL, 10.0.2.2 为模拟器本地的地址
        // 第 1 步: 创建 SoapObject 对象
        SoapObject req= new SoapObject(SPACE,
        METHOD);
        // 第 2 步: 设置 WebService 方法的参数值
        request.addProperty("product", "computer");
        // 第 3 步: 创建 SoapSerializationEnvelope 对象
        SoapSerializationEnvelope envelope = new
        SoapSerializationEnvelope(SoapEnvelope.VER11);
  
```

```

envelope.dotNettrue;
envelope.setOutputSoapObject(req);
//第 4 步: 创建 HttpTransportSE 对象
AndroidHttpTransport androidHttpTransport = new
AndroidHttpTransport(URL);
//第 5 步:返回结果
Object result = (Object)envelope.getResponse();
} catch (Exception e) {……//出错处理 }
//使用 UI 组件 viewResult 的 post 方法更新结果
viewResult .post(new Runnable(){
……//重写 run()方法 });
return null;} }
@Override
public void onClick(View view){
// 执行异步调用 Webservice 的任务
new MyTask ().execute();}
@Override
public void onCreate(Bundle savedInstanceState)
{
……//设置页面布局, 获取组件等相关操作
//为按钮 btnSearch 组件添加点击监听事件
btnSearch.setOnClickListener(this);} }

```

(3) 移动终端中 BI 图表的设计

目前 Android 系统主要有 Java4Less、Chart4J、BlueChart、aiCharts 和 aChartEngine 等图表库, 本例采用 aChartEngine。该图表工具库专为 Android 应用而设计, 目前支持曲线图、柱状图、条形图、饼图和散点图等多种图表格式, 并具有图表缩放功能。每种图表可以有多个序列, 支持多种自定义特性, 并允许将 X 轴以水平或垂直的方式显示。此外, 图表还可以作为 View 或 Intent 加以构建, 用于启动一个活动。移动终端 BI 图表设计的核心程序代码如下。

```

//XML 文档解析器
XmlPullParser parser = Xml.newPullParser();
//获取事件类型
int eventCode = parser.getEventType();
while(eventCode!= XmlPullParser.END_
DOCUMENT){
switch(eventCode){ ……//根据事件类型把结果增
加到 results 结果集中}
eventCode=parser.next(); //解析下个元素}

```

4 系统建设 System Construction

```

//描绘器 getRender()方法用来设置曲线图的格式,
包括颜色、值的范围、点和线的形状等
private XYMultipleSeriesRenderer mRd
=getRender();
//数据设置器, 为图表进行赋值
XYMultipleSeriesDataset dataset = new
XYMultipleSeriesDataset();
CategorySeries series = new CategorySeries( "本年
销售"); //声明柱形图
//根据结果集的值柱形图添加数据
for(int i = 0;i<results.size();i++){
//把每条记录转换成对应的实体 c 类
series.add(((Sale)list.get(i)).get("money"));
//添加该柱形图到数据设置列表
dataset.addSeries(series.toXYSeries());
//X 轴上的刻度标签}
mRd.addTextLabel(i,((Sale)list.get(i)).get("month"))
;
Intent intent = ChartFactory.getBarChartIntent ( this,
dataset, mRd, Type.DEFAULT); //为这个 Activity 添
加 Intent
startActivity(intent);

```

4 结语

移动商业智能是目前企业信息化领域的一个崭新的研究课题, 传统的商业智能已面临较大的挑战。随着知识管理技术、数据分析与挖掘技术、移动通信技术的不断创新, 实时商业智能、移动商业智能等将成为企业商业智能系统应用的主要趋势之一。本文提出了集成 ERP 的移动商业智能概念, 详细设计了一个具有可扩展性、低耦合性、实用性和跨平台性的移动商业智能系统架构, 并运用 Android 和 Web 服务等工具对其实现技术进行了详细描述。随着智能移动终端设备及其操作系统的不断升级, 下阶段工作将对不同操作系统下客户端 BI 图表工具库进行分析比较, 对企业应用系统与移动终端 BI 系统之间的即时通信技术进行深入研究。

参考文献

- 1 周瑾, 黄立平. 知识管理与商业智能关系研究. 科学学与科学技术管理, 2009, 3: 100-105.

(下转第 13 页)